LSI 03-2300

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR PATENT

ON

METHOD AND APPARATUS FOR PERFORMING LOGICAL TRANSFORMATIONS FOR GLOBAL ROUTING

BY

ALEXEI V. GALATENKO
LITOVSKIY BULVAR
D. 3, KOR. 2, KV. 351
MOSCOW, RUSSIA 117593
CITIZEN OF RUSSIA

ELYAR E. GASANOV
ULITSA GRINA
D. 40, KV. 24
MOSCOW, RUSSIA 117628
CITIZEN OF RUSSIA

ANDREJ A. ZOLOTYKH
ULITSA INSTITUTSKAYA, D. 12, KV. 25
FRYAZINO, MOSKOVSKAYA OBLAST
RUSSIA 141120
CITIZEN OF RUSSIA

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"Express Mail" Mailing Label Number: EV 380 207 234 US

Date of Deposit: March 17, 2004

I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on the date indicated above and is addressed to Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

BY: Penny L. Flint
Penny L. Flint

METHOD AND APPARATUS FOR PERFORMING LOGICAL TRANSFORMATIONS FOR GLOBAL ROUTING

CROSS-REFERENCE TO RELATED DOCUMENTS

[0001] The present application herein incorporates the following U.S. Patents by reference in their entirety: (1) United States Patent No. 6,564,361, entitled *Method and Apparatus for Timing Driven Resynthesis*, issued to A. Zolotykh et al. on May 13, 2003; and (2) United States Patent No. 6,543,032, entitled *Method and Apparatus for Local Resynthesis of Logic Trees with Multiple Cost Functions*, issued to A. Zolotykh et al. on April 1, 2003.

FIELD OF THE INVENTION

[0002] The present invention generally relates to the field of integrated circuits, and particularly to a method and apparatus for performing logical transformations for global routing.

BACKGROUND OF THE INVENTION

[0003] A typical integrated circuit (IC) design flow often includes a logical design and a physical design. A logic design may include the following steps: (1) design entry (to enter the design into an IC design system, either using a hardware description language or schematic entry); (2) logical synthesis (to use a hardware description language such as VHDL or Verilog or the like and a logical synthesis tool to produce a netlist, which is a description of logical cells and their connections); (3) system partitioning (to divide a large system into small pieces); and (4) pre-layout simulation (to check whether a design functions correctly). A physical design may include the following steps: (1) floorplanning (to arrange blocks of a netlist on a chip); (2) placement (to decide the location of cells in a block); (3) routing (to make connections between cells and blocks); (4) extraction (to determine resistance and capacitance of the interconnect); and (5) post-

layout simulation (to check whether the design still works with the added loads of the interconnect).

[0004] Once the IC designer has floorplanned a chip and logic cells within the flexible blocks have been placed, it is time to make the connections by routing the chip. Routing is usually divided into global routing and detailed routing. In global routing, connections are completed between the proper blocks of the IC, disregarding the exact geometric details of each wire and terminal. For each wire, a global router finds a list of channels that are to be used as a passageway for that wire. In other words, global routing specifies the loose route of a wire through different regions of the routing space. Thus, a global router does not make any connections – it just plans them. Global routing is followed by detailed routing which completes point-to-point connections between terminals on the blocks. Loose routing is converted into exact routing by specifying the geometric information such as width of wires and their layer assignments. Detailed routing includes the exact channel routing of wires.

[0005] The input to a global router is a floorplan that includes the location of all fixed and flexible blocks, the placement information for flexible blocks, and the location of all logical cells. The goal of global routing is to provide complete instructions to the detailed router on where to route every net. Global routing tries to achieve at least one of the following objectives: (1) to minimize the total interconnect length; (2) to maximize the probability that detailed router can complete the routing; and (3) to minimize the critical path delay. Many methods used in global routing are based on the solution to the tree on a graph problem. Therefore, it would be desirable to provide a method and apparatus for performing logical transformations for global routing to optimizing design parameters.

SUMMARY OF THE INVENTION

[0006] Accordingly, the present invention provides a new approach and algorithm to optimize various design parameters in global routing. According to an exemplary aspect of the present invention, marked trees are first preprocessed. For every vertex incident to leaves, one may go through the list of its leaves, and if two leaves have the same mark one may leave only one of them. After that whether homeomorphism exists may be determined. The reason behind selecting such homeomorphic pairs is as follows: adding or removing a vertex of degree 2 as well as adding or removing a new leaf (variable) does not significantly modify routing (in this case all routing transformations are in essence splitting and merging routing trees). After the selection of applicable transformations, one may apply them to optimize design parameters. This may be achieved by assigning the same coordinates to nodes of degree != 2 of homeomorphic trees, which means that one may assign the coordinates of corresponding nodes to "essential" nodes and then insert or remove nodes of degree 2.

[0007] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as claimed. The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate an embodiment of the invention and together with the general description, serve to explain the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The numerous advantages of the present invention may be better understood by those skilled in the art by reference to the accompanying figures in which:

FIG. 1 illustrates homeomorphic trees in accordance with an exemplary embodiment of the present invention;

FIG. 2 illustrates trees that are not homeomorphic in accordance with an exemplary embodiment of the present invention;

FIGS. 3A and 3B depict a flowchart showing a method for determining homeomorphism between two logical trees A and B in accordance with an exemplary embodiment of the present invention;

FIG. 4 is a flowchart showing a method for assigning coordinates to nodes of homeomorphic trees A and B in accordance with an exemplary embodiment of the present invention;

FIG. 5 shows an original path to assign coordinates in accordance with an exemplary embodiment of the present invention; and

FIG. 6 shows placing essential nodes in accordance with an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0009] Reference will now be made in detail to the presently preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings.

[0010] The present invention provides a new approach and algorithm to optimize various design parameters in global routing. The algorithm may be used to generate a list of transformation families which may be utilized in the process of design resynthesis without significant routing modifications. The concrete transformations may then be utilized to improve design parameters.

A. Introduction

[0011] A transformation family is a pair of marked logical trees. A pair of trees is called suitable if and only if the trees are homeomorphic. Two trees T_1 and T_2 are homeomorphic if and only if T'_1 and T'_2 are isomorphic, where T'_1 and T'_2 are obtained from T_1 and T_2 by removing all vertices of degree 2, respectively. Two trees are isomorphic if and only if there exists a one-to-one correspondence between their vertices and edges while preserving the incidence relationship.

[0012] A tree is a graph in which any two vertices are connected by exactly one path. A path is a sequence of vertices such that from each of its vertices there is an edge to the successor vertex. An edge connects two vertices; these two vertices are said to be incident to the edge. The degree of a vertex is the number of edges incident to the vertex. A vertex is also called a node. A child node is a node that is linked to by a parent node. A leaf node (or leaf) is a node with no child nodes.

[0013] FIG. 1 illustrates homeomorphic trees 100 in accordance with an exemplary embodiment of the present invention. As shown, the tree 102 has a vertex set V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}and an edge set E= {{1,2}, {2,3}, {3,4}, {3,7}, {3,9}, {4,5}, {5,6}, {7,8}, {9,10}}. The unique simple path connecting the vertices 6 and 10 is 6-5-4-3-9-10. The vertices 1, 6, 8, and 10 have a degree of 1, the vertices 2, 4, 5, 7, and 9 have a degree of 2, and the vertex 3 has a degree of 4. Those of ordinary skill in the art will understand that the trees 102 and 104 are homeomorphic. FIG. 2 illustrates trees that are not homeomorphic in accordance with an exemplary embodiment of the present invention.

[0014] According to one aspect of the present invention, marked trees are first preprocessed. For every vertex incident to leaves, one may go through the list of its leaves, and if two leaves have the same mark one may leave only one of them. After that one may look for homeomorphism. The reason behind selecting such homeomorphic pairs is as follows: adding or removing a vertex of degree 2 as well as adding or removing a new leaf (variable) does not significantly modify routing (in this case all routing transformations are in essence splitting and merging routing trees).

[0015] After the selection of applicable transformations, one may apply them to optimize design parameters. This may be achieved by assigning the same coordinates to nodes of degree != 2 of homeomorphic trees, which means that one may assign the coordinates of corresponding nodes to "essential" nodes and then insert or remove nodes of degree 2.

[0016] Let A and B be two logical trees. Some nodes of these trees may be marked. If a node is marked, it means that a sub-tree with a root node in this node may not change

when B is obtained from A. One may call a marked node of A corresponding to a marked node of B, if these nodes have the same coordinates and the same type.

B. Determining Homeomorphism

[0017] FIGS. 3A and 3B depict a flowchart showing a method for determining homeomorphism between two logical trees A and B in accordance with an exemplary embodiment of the present invention. As shown, the method may include the following steps.

1) Step A

[0018] Pairs (Na, Nb) may be found, where Na is a marked node of A, Nb is a marked node of B, and Na is a corresponding node of Nb. For the n-th pair (Na, Nb), one may assign a value n to the parameter fixed_corr of Na and Nb. For all nodes that are not in pairs of corresponding nodes, one may assign NO_FIXED_CORR value to fixed_corr. If there are marked nodes in B for which no corresponding nodes in A exist, an error code may be returned.

2) Step B

[0019] A code (a pair (I, a_1 a_2 ... a_I), where I is the length and a_1 ... a_I is an array of integers) may be assigned to each node of A and B. The following procedure may be used to assign a code to a node N:

i) If N is a variable, the code is (1, variable number);

ii) If N is a constant, the code is (1, -1); and

iii) Suppose that N has children and codes of children are determined as follows:

a) If all children have the code (1, -l), the same code (1, -l) is assigned to N; and

b) (1) If there are several children that are variables with

the same variable number, one may leave only one of these children;

(2) If only one significant (i.e., with a code different from (1, -1)) child is left after (1), and this child has a code (I, a_1 ... a_I), one may assign the same code to N if N.fixed_corr == NO_FIXED_CORR; otherwise, one may assign code (I+2, 1 N.fixed_corr a_1 ... a-I) to N; and

(3) If several significant children are left after (1), and codes of these children are (I_1, c_1) ... (I_k, c_k) (k >= 2) (c_j = a_j,1 ... a_j,c_j), one may order these codes lexicographically (c_j(1) <= c_j(2) <= ... <= c_j(k)) and assign code (I_1 + ... + I_k + 2, k N.fixed_corr I_j(1) ... I_j(k) c_j(1) ... c_j(k)) to N.

3) Step C

[0020] If root nodes have the same code, homeomorphism exists. Otherwise, homeomorphism cannot be established.

C. Assigning Coordinates

[0021] FIG. 4 is a flowchart showing a method for assigning coordinates to nodes of homeomorphic trees A and B in accordance with an exemplary embodiment of the present invention. As shown, the method may include the following steps.

1) Step A

[0022] All space occupied by the logical tree A may be cleared.

2) Step B

[0023] One may assign corr_node fields for nodes of A and B with (number of children != 1) or (fixed_corr != NO_FIXED_CORR) in the following way: corr_nodes should have the same code. If a node has several children that are variables with the same variable number, one may assign corr_node only for one of these variables; for others, one may put corr_node = NULL. For nodes with fixed_corr = NO_FIXED_CORR and children number = 1, one may assign corr_node = NULL. If one fails to assign corr_node to some nodes, an error code is returned.

3) Step C

[0024] One may assign coordinates to nodes N of B with the number of children >= 2 or with fixed_corr != NO_FIXED_CORR (the coordinates are the coordinates of N-> corr_node). One may also assign a routing tree to N.corr_node (one may merge the routing tree of N.corr_node with routing trees of father, grandfather, ... of N.corr_node while these grandfathers have only one child). One may change free space to show that the node is added to the field.

4) Step D

[0025] One may slightly change correspondence (corr_nodes) for variables to achieve the following: routing positions of corresponding variables should coincide. For variables with NULL corr_node in B, one may put routing position = -1.

5) Step E

[0026] One may assign variable trees as follows: for each variable, one may merge the tree of these variables with routing trees of all nodes from A that have the only child -- this variable.

6) Step F

[0027] If a node of A has several children that are variables with the same variable number, one may delete parts of this variable's routing tree that corresponds to these

children except one (the one that does not have NULL corr_node, see the foregoing Step D) (see, e.g., FIGS. 5 and 6).

7) Step G

[0028] One may insert nodes N of B with the number of children = 1. This may be accomplished by trying to insert this node to every segment between the first grandfather and the first grandson that have 0 or >= 2 children, and determine if this is possible (i.e. if there is place left). If successful, one may split the routing tree of grandchild, assign the routing tree to N (the second part of split tree), and change free space; if one fails to insert a node, an error code is returned and one may go to Step J.

8) Step H

[0029] One may assign routing positions to variables that have routing position = -1. This may be accomplished by adding the imaginary segment to the variable routing tree.

9) Step I

[0030] One may re-compute delays and caps.

10) Step J

[0031] One may clear the space occupied by B (by nodes that one has managed to put), put back nodes of A, and return the result: an error code or a success code. If the result is a success code, all nodes of B have correct coordinates and routing trees assigned.

[0032] It is to be noted that the above described embodiments according to the present invention may be conveniently implemented using conventional general purpose digital computers programmed according to the teachings of the present specification, as will be apparent to those skilled in the computer art. Appropriate software coding may readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art.

[0033] It is to be understood that the present invention may be conveniently implemented in forms of software package. Such a software package may be a computer program product which employs a storage medium including stored computer code which is used to program a computer to perform the disclosed function and process of the present invention. The storage medium may include, but is not limited to, any type of conventional floppy disks, optical disks, CD-ROMS, magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, or any other suitable media for storing electronic instructions.

[0034] It is understood that the specific order or hierarchy of steps in the processes disclosed is an example of exemplary approaches. Based upon design preferences, it is understood that the specific order or hierarchy of steps in the processes may be rearranged while remaining within the scope of the present invention. The accompanying method claims present elements of the various steps in a sample order, and are not meant to be limited to the specific order or hierarchy presented.

[0035] It is believed that the present invention and many of its attendant advantages will be understood by the foregoing description. It is also believed that it will be apparent that various changes may be made in the form, construction and arrangement of the components thereof without departing from the scope and spirit of the invention or without sacrificing all of its material advantages. The form herein before described being merely an explanatory embodiment thereof, it is the intention of the following claims to encompass and include such changes.